Hubless Nearest Neighbor Search for Bilingual Lexicon Induction

Jiaji Huang* **Qiang Qiu†** **Kenneth Church***
* Baidu Research, Sunnyvale, CA, 94089
† Department of Electrical Engineering, Duke University, Durham, NC, 27707
*{huangjiaji, kennethchurch}@baidu.com †qiang.qiu@duke.edu

Abstract

Bilingual Lexicon Induction (BLI) is the task of translating words from corpora in two languages. Recent advances in BLI work by aligning the two word embedding spaces. Following that, a key step is to retrieve the nearest neighbor (NN) in the target space given the source word. However, a phenomenon called hubness often degrades the accuracy of NN. Hubness appears as some data points, called hubs, being extra-ordinarily close to many of the other data points. Reducing hubness is necessary for retrieval tasks. One successful example is Inverted SoFtmax (ISF), recently proposed to improve NN. This work proposes a new method, Hubless Nearest Neighbor (HNN), to mitigate hubness. HNN differs from NN by imposing an additional *equal preference assumption*. Moreover, the HNN formulation explains why ISF works as well as it does. Empirical results demonstrate that HNN outperforms NN, ISF and other state-of-the-art. For reproducibility and follow-ups, we have published all code[1].

1 Introduction

This paper presents a new method for Bilingual Lexicon Induction (BLI), which we call *Hubless Nearest Neighbor* (HNN). BLI is the task of creating a lexicon of translation equivalents such as, *bank:banc* or *bank:banque* automatically from non-parallel corpora. The proposed method not only improves upon but also unifies several recent works that retrieve translations by *Nearest Neighbor* (NN) (Mikolov et al., 2013) and more advanced techniques like *Inverted SoFtmax* (ISF) (Smith et al., 2017).

There is a long history of BLI using non-parallel corpora. Methods often rely on some designed features, which reveal some shared structures across languages. For example, co-occurrence matrices (Rapp, 1995), tf-idf of context words (Fung, 1998), and lexical similarities (Klementiev et al., 2012). A comprehensive study can be found in (Irvine and Callison-Burch, 2017).

Recently, Mikolov et al. (2013) observe that isomorphism exists across word embeddings of different languages. This motivates them to learn a linear mapping to align the spaces, using a seeding dictionary of 5K pairs of translations. After that, more translations can be induced by NN search. Following the seminal work, significant advances have been made. For example, Faruqui and Dyer (2014) use Canonical Component Analysis to align the two embedding spaces. Xing et al. (2015) show a substantial gain by normalizing the embeddings and constraining the mapping to be orthogonal. A series of works by Artetxe et al. (2017, 2018a,b) show that decent accuracies can be achieved even with a tiny or no seeding dictionary. The authors name their method as "self-learning", which alternates between learning the mapping and inducing more translation pairs. The similar methodology is also seen in (Zhang et al., 2017b), where the induction step reduces a cost called earth mover distance. Conneau et al. (2018) propose to use Generative Adversarial Network (Goodfellow et al., 2014) to learn the mapping when no seeding dictionary is available.

Whether using a seeding dictionary or not, the induction always requires to retrieve the translation under some distance measure. NN may be the most straightforward approach. However, it is often challenged by a phenomenon called hubness (Radovanovic et al., 2010). Hubness is a tendency that a few words (hubs) are too near to too many other words, especially in high dimensional spaces. It degrades the accuracy of NN in various tasks (Aucouturier and Pachet, 2008; Ozaki et al.,

[1] https://github.com/baidu-research/HNN

2011; Suzuki et al., 2013; Zhang et al., 2017a), including BLI (Dinu et al., 2014). Recently, remarkable improvements have been made in BLI by mitigating hubness. For example, Smith et al. (2017) propose Inverted SoFtmax (ISF) that scales the similarities by a (global) measure of hubness. Conneau et al. (2018) develop a method called Cross domain Similarity Local Scaling (CSLS) that relies on a local measure of hubness instead.

This work studies how to overcome hubness in BLI. The new method, HNN, is proposed by introducing an *equal preference assumption*. As we shall see, the assumption leads to an optimization problem that manifests the connection between HNN, NN and ISF. Empirical results demonstrate that HNN is very competitive and able to outperform NN, ISF and other state-of-the-art like CSLS. In summary, the paper makes the following contributions:

1. We propose an optimization based framework that connects NN, ISF and the proposed HNN.
2. We derive an efficient solver for HNN, which outperforms NN, ISF and other state-of-the-art.
3. We show that ISF is a part of HNN's solver, which explains why ISF works.

2 Bilingual Lexicon Induction with a Seeding Dictionary

Since hubness is the major concern of this work, we focus on the case with a seeding dictionary for simplicity. Representative methods (Mikolov et al., 2013; Xing et al., 2015; Artetxe et al., 2018a) often consist of two steps: 1) learning a mapping that aligns the source and target embedding spaces; 2) given a source word, retrieve according to some distance metric, in the target embedding space. We briefly review the two steps in this section.

Let the source word embeddings space be $\mathcal{X} \subset \mathbb{R}^d$, and the target space be $\mathcal{Y} \subset \mathbb{R}^d$. A typical value of the dimension d is 300. Suppose the vocabulary sizes of source/target languages are m and n respectively. Then $\mathcal{X}$ is a set of m embedding vectors, denoted as

$$\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\},$$

and $\mathcal{Y}$ is a set of n embeddings,

$$\mathcal{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}.$$

2.1 Learning Linear Transformation

Suppose we can access a seeding dictionary, which reveals some correspondences from a subset of $\mathcal{X}$ to a subset of $\mathcal{Y}$. The correspondence can be one-to-one, many-to-one, or one-to-many. In matrix form, let the columns of matrix $\mathbf{X}$ ($\mathbf{Y}$) be the embeddings of source (target) words in the dictionary, where the j-th columns of $\mathbf{X}$ and $\mathbf{Y}$ are the embeddings of a pair of translations.

Using $\mathbf{X}$ and $\mathbf{Y}$, a linear mapping $\mathbf{T}$ can be learned, which "aligns" the $\mathcal{X}$ and $\mathcal{Y}$ space. In particular,

$$\mathbf{T} = \arg\min_{\mathbf{T} \in \Omega} \|\mathbf{T}\mathbf{X} - \mathbf{Y}\|_F^2.$$

Here Ω is a constraint set on $\mathbf{T}$. For example, Mikolov et al. (2013) simply let Ω be $\mathbb{R}^{d \times d}$, whereas Xing et al. (2015) show substantial gain by setting Ω as the set of orthogonal matrices.

2.2 Retrieval by Nearest Neighbor (NN)

Once the $\mathbf{T}$ is obtained, translation can be cast as a retrieval problem. We define a distance matrix $\mathbf{D}$, between the mapped source embeddings and target embeddings,

$$D_{i,j} \triangleq \text{dist}(\mathbf{T}\mathbf{x}_i, \mathbf{y}_j),$$

where "dist" is some distance metric. For the i-th source word, Nearest Neighbor (NN) criterion determines (the index of) its translation in the $\mathcal{Y}$ set, by

$$\arg\min_j D_{i,j} \qquad \text{(NN)}$$

However, several works (Radovanovic et al., 2010; Dinu et al., 2014) have observed that the accuracy of NN is often significantly degraded by a phenomenon called hubness. Mitigating hubness has thus become necessary, which we will review next.

2.3 Inverted Softmax: Improve NN by Mitigating Hubness

Hubness occurs in high dimensional feature space (Radovanovic et al., 2010). It appears as some data points, called hubs, being close to too many of the others. We look into Inverted SoFtmax (ISF) (Smith et al., 2017), a recent retrieval methods that tackle hubness.

Given the distance matrix $\mathbf{D}$, ISF seeks a matrix ISF where its i, j-th entry decides the probability of translating the i-th source word to the

j-th target. A "temperature" parameter ϵ is introduced to construct a kernel, $\exp(-D_{i,j}/\epsilon)$. The ISF matrix is obtained by normalizing the kernel's columns first, and then the rows.

$$ISF(\mathbf{D})_{i,j} = \frac{\exp\left(-D_{i,j}/\epsilon\right)}{Z_i \sum_{i=1}^{m} \exp\left(-D_{i,j}/\epsilon\right)},$$

where Z_i is a row normalizer such that $\sum_j ISF_{i,j} = 1$ for any i. The (index of) translation for the i-th source word is then determined by

$$\arg\max_j ISF(\mathbf{D})_{i,j}. \qquad \text{(ISF)}$$

Smith et al. (2017) show that ISF significantly outperforms NN in BLI tasks. However, it is still not clear why ISF works so well. One contribution of our work is to shed light on the mechanism behind ISF, which will be manifested after we study the proposed Hubless Nearest Neighbor.

3 Hubless Nearest Neighbor (HNN)

This section formalizes the proposed HNN. As will be clear from this section and the next, there is a unified view over NN, ISF and HNN. We now start by rephrasing the retrieval task into the following general problem. Given the distance matrix $\mathbf{D}$ defined in section 2.2, we seek an assignment matrix $\mathbf{P}$ where $P_{i,j}$ is the probability of the j-th target word being the translation of the i-th source word. Assume the target vocabulary is large enough so that one or more translations can always be found for any source word. Therefore

$$\sum_j P_{i,j} = 1.$$

The (index of) translation is inferred by

$$\arg\max_j P_{i,j}.$$

The art is to determine $\mathbf{P}$ from $\mathbf{D}$ and various other information/priors. The above framework is general in the sense that various designs exist in seeking the $\mathbf{P}$.

3.1 NN as a Warm-up

As a warm-up, we show that NN is a special case of the above framework. In specific, let $\langle\cdot,\cdot\rangle$ be matrix inner product, then $\langle\mathbf{D},\mathbf{P}\rangle$ is a measure of cost to translate from source to target, which we may want to minimize. In addition, if we minimize it along with a negative entropy regularizer (on $\mathbf{P}$), we can reduce the gap between NN and (ISF). As stated by the following proposition,

Proposition 1. *The* (NN) *criterion is equivalent to* $\arg\max_j P_{i,j}$*, where* $\mathbf{P}$ *is the solution of the following optimization problem,*

$$\begin{aligned} &\min_{\mathbf{P}} \langle\mathbf{D},\mathbf{P}\rangle + \epsilon \sum_{i,j} P_{i,j} \log P_{i,j} \\ &s.t. P_{i,j} \geq 0,\ \sum_j P_{i,j} = 1 \end{aligned} . \qquad (\mathcal{P}_0)$$

Proof. The solution to ($\mathcal{P}_0$) is simply

$$P_{i,j} = \frac{\exp(-D_{i,j}/\epsilon)}{\sum_{j=1}^{n} \exp(-D_{i,j}/\epsilon)}. \qquad (1)$$

Substituting Eq. (1) to $\arg\max_j P_{i,j}$, we arrive at

$$\arg\min D_{i,j},$$

which is exactly the NN criterion. □

The objective of ($\mathcal{P}_0$) is regularized by $\epsilon \sum_{i,j} P_{i,j} \log(P_{i,j})$, which is the negative entropy of $\mathbf{P}$. It smooths the linear objective, and leads to a solution, Eq. (1), as anther view of NN that is closer to (ISF).

3.2 Equal Preference Assumption

Starting from ($\mathcal{P}_0$), we now try to reduce the hubs. Since hubness results in some y_j's being retrieved more frequently than others, a natural idea is to force all y_j's being equally preferred to be retrieved. The preference of y_j can be measured by

$$pf_j(\mathbf{P}) \triangleq \frac{1}{m} \sum_i P_{i,j}.$$

In other words, on average, how the j-th target word is likely to be picked as the translation of a source word. We therefore force $\frac{1}{m}\sum_i P_{i,j}$ to be uniform over all j's. In addition, with the constraint that $\sum_j P_{i,j} = 1$ for any i, we have $\frac{1}{m}\sum_i P_{i,j} = \frac{1}{n}$ for any j. We term the constraint as equal preference assumption. Formally,

Definition 1. *Equal Preference Assumption:*

$$pf_j = \frac{1}{n},$$

for all j's.

If the translation is strictly one-to-one, then $m = n$ and $\mathbf{P}$ is a permutation matrix. The assumption exactly holds. In reality, translation is not one-to-one. But empirically, we still observe that it approximately holds, at least for some language pairs. To see that, we build a "groundtruth"

$\mathbf{P}^*$ using an English-French dictionary[2]. The dictionary includes 113K items, with plenty of polysemies. The vocabularies are built from monolingual word embedding files [3]. Vocabulary sizes are set as $m = n = 500,000$. The $\mathbf{P}^*$ is computed as

$$P^*_{i,j} \triangleq \frac{1}{Z_i} \cdot \begin{cases} 1 & i\text{-th source word can be} \\ & \text{translated to the } j\text{-th target} \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

where Z_i is a normalizer that ensures $\sum_j P^*_{i,j} = 1$. We compute the pf_j values using the $\mathbf{P}^*$. To measure how much the values deviate from the equal preference assumption, we compute their variance,

$$\text{var}_j[pf_j]. \tag{3}$$

If the equal preference assumption exactly holds, $\text{var}_j[pf_j] = 0$. Otherwise, the larger the variance, the less true the assumption is. It turns out that $\text{var}_j[pf_j] \approx 1.9 \times 10^{-11}$, which is tiny and support the assumption.

3.3 HNN

We add the equal preference assumption as a constraint to problem ($\mathcal{P}_0$), and now solve the following new problem instead,

$$\begin{aligned} &\min_{\mathbf{P}} \langle \mathbf{D}, \mathbf{P} \rangle + \epsilon \sum_{i,j} P_{i,j} \log P_{i,j} \\ &s.t. P_{i,j} \geq 0, \ \sum_j P_{i,j} = 1, \ \frac{1}{m} \sum_i P_{i,j} = \frac{1}{n}. \end{aligned} \tag{$\mathcal{P}_1$}$$

In analogous to (NN) and Proposition 1, we introduce the following definition of Hubless Nearest Neighbor (HNN).

Definition 2. *HNN is the criterion that retrieves index*

$$\arg\max_j P_{i,j}$$

where $\mathbf{P}$ *is the solution of problem* ($\mathcal{P}_1$).

The remaining question is how to solve ($\mathcal{P}_1$), which we will discuss in the next section.

4 Solvers for HNN

We first present a straightforward but less efficient solver, then we derive an efficient alternative. Both solvers, as will be seen, have strong connections with ISF.

[2] https://dl.fbaipublicfiles.com/arrival/dictionaries/en-fr.txt

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.en.vec and https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.fr.vec

4.1 A Less Efficient Solver

($\mathcal{P}_1$) can be solved by Sinkhorn iteration (Cuturi, 2013), which iteratively normalizes the columns and rows of a kernel matrix. The steps are summarized in algorithm 1.

Algorithm 1 Sinkhorn Solver for Problem ($\mathcal{P}_1$)

Input: $\mathbf{D}$
Output: $\mathbf{P}$

$\mathbf{P} \leftarrow \exp(-\mathbf{D}/\epsilon)$ where exp is on elements.
while stopping criteria not met **do**
  // normalize columns
  $\mathbf{P} \leftarrow \mathbf{P}\text{diag}\left\{\frac{m}{n}./(\mathbf{P}^\top \mathbf{1})\right\}$
  // normalize rows
  $\mathbf{P} \leftarrow \text{diag}\{1./(\mathbf{P}\mathbf{1})\}\mathbf{P}$
end while

Here ./ denotes elementwise division, $\mathbf{1}$ is an all-one vector of suitable length, and $\text{diag}\{\cdot\}$ constructs a diagonal matrix from a vector.

Remark 1. *The Sinkhorn Solver reveals some connection between ISF and HNN. Indeed, ISF is equivalent to running a single iteration of algorithm 1. A deeper connection will be revealed in the next subsection, where we provide a complementary view of problem* ($\mathcal{P}_1$).

4.2 Dual Problem and an Efficient Solver

One drawback of algorithm 1 is its prohibitive memory cost. Indeed, the $\mathbf{P}$ matrix has to be in memory for frequent update, which is costly when the vocabulary sizes m and n are big. In this section, we study a dual form of problem ($\mathcal{P}_1$), given in Proposition 2, which hints a more efficient method to solve for $\mathbf{P}$.

Proposition 2. *The solution of problem* ($\mathcal{P}_1$) *can be expressed as*

$$P_{i,j} = \frac{\exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right)}{\sum_j \exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right)}, \tag{4}$$

where β_j *is the solution of*

$$\min_{\boldsymbol{\beta}} \frac{1}{m} \sum_i \left[\epsilon \log \sum_j \exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right) \right] - \frac{1}{n} \sum_j \beta_j \tag{$\mathcal{D}$}$$

Proof. The proof is by method of Lagrangian multipliers. Details are in supplementary. □

The dual form ($\mathcal{D}$) is a special case of the dual forms in general contexts (Genevay et al., 2016). ($\mathcal{D}$) is useful because it allows us to learn a much smaller vector $\boldsymbol{\beta}$ (of size n), instead of keeping updating the huge matrix $\mathbf{P}$ (m by n). Moreover, its loss function is a summation over the i's, which can be minimized by parallelizable gradient descent. Finally, ($\mathcal{D}$) is convex, which guarantees the convergence of gradient descent.

It is now a natural idea to derive the gradient of ($\mathcal{D}$) and give a gradient descent solver. Let us define the objective in ($\mathcal{D}$) as ℓ_i,

$$\ell_i \triangleq \epsilon \log \sum_j \exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right) - \frac{1}{n}\sum_j \beta_j.$$

Then,

$$\frac{\partial \ell_i}{\partial \beta_j} = \frac{\exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right)}{\sum_j \exp\left(\frac{\beta_j - D_{i,j}}{\epsilon}\right)} - \frac{1}{n}. \tag{5}$$

Algorithm 2 summarizes the gradient descent solver. The equivalence between algorithm 1 and 2 will be empirically validated in section 5.1. In large-scale experiments, we apply algorithm 2 (instead of algorithm 1) for its lower memory cost.

Algorithm 2 Dual Solver for Problem ($\mathcal{P}_1$)

Input: $\mathbf{D}$, learning rate η.
Output: $\mathbf{P}$

$\boldsymbol{\beta} \leftarrow \mathbf{0}$
while stopping criteria not met **do**
  for $i = 1, \ldots, m$ **do**
    // parallelizable
    Compute gradient $\nabla \ell_i$ using E.q. (5).
  end for
  $\boldsymbol{\beta} \leftarrow \boldsymbol{\beta} - \eta \cdot \frac{1}{m}\sum_i \nabla \ell_i$
end while
Compute $\mathbf{P}$ by E.q. (4) and return.

Remark 2 (Unifying NN, ISF and HNN). *Comparing the* $\mathbf{P}$ *matrix under NN (Eq.* (1)*) and HNN criterion (Eq.* (4)*), we observe that HNN introduces an additional set of values,* β_j*'s. The quantity,* $\exp(-\beta_j/\epsilon)$*, normalizes the* j*-th column of the kernel matrix* $\exp(-D_{i,j}/\epsilon)$*. In contrast,* (ISF) *simply sets the column normalizer as the column sum,* $\sum_i \exp(-D_{i,j}/\epsilon)$*. Obviously, HNN works harder in figuring out the normalizers, which results in a higher accuracy, as we shall see in the experiment section.*

5 Experiments

In this section, we first experiment with synthetic data to illustrate the connection between NN, ISF and the proposed HNN. Then, we report extensive results on BLI task to show the advantage of HNN over NN, ISF and CSLS, another state-of-the-art.

We will also demonstrate that hubness is indeed reduced by HNN. To measure hubness, we adopt the *k-occurrence* metric proposed in (Radovanovic et al., 2010), but with a small adaption. In its original definition, k-occurrence, N_k, is the number of times a data point appears among the k nearest neighbors of all the others. In our case, we measure for every target example, the number of times it is retrieved against the source set. If a target example is retrieved too many times, it is likely to be a "hub". For both the original definition and our adapted one, hubness can be indicated by a long tail of the distribution of N_k.

5.1 Connection between NN, ISF and HNN

We simulate a retrieval task, where source and target spaces are already aligned. In specific, data is generated from a Gaussian mixture model,

$$\sum_{c=1}^{10,000} \frac{1}{10,000} \cdot \mathcal{N}(\boldsymbol{\mu}_c,\ 0.01 \times \mathbf{I}_d),$$

where the dimension $d = 300$. The class mean $\boldsymbol{\mu}_c$ is generated by normalizing a $\mathbb{R}^{300}$ vector where each dimension is drawn from uniform$(-1, 1)$. We generate two samples per class, one in the source set, the other as the target to be retrieved. We use NN, ISF and HNN with algorithm 1 and 2 to retrieve for the 10K source samples. ϵ is set to 0.1, which gives the best accuracy of ISF. The same ϵ is also used in HNN.

Table 1 reports top-1, top-5, top-10 accuracies. The two algorithms for HNN achieve the best results and their accuracies are basically the same, validating their equivalence. To understand the improvement over NN, we measure the hubness in different methods. Figure 1 plots the distribution of N_1, the 1-occurrences. HNN has the shortest tail, in stark contrast to the long tail of NN, implying significantly reduced number of hubs.

We then illustrate the connection between ISF and HNN. Figure 2 tracks the top-1 accuracies

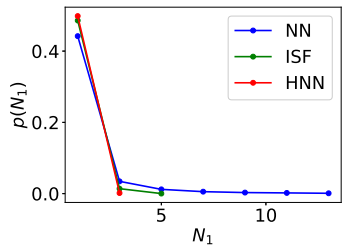


Figure 1: Distribution of N_1 (1-occurrences): the number of times a target example is retrieved. Longer tail indicates more hubs.

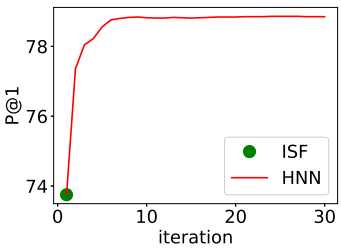


Figure 2: P@1 over the iterations in algorithm 1. ISF's P@1 overlaps with that of HNN at the 1st iteration.

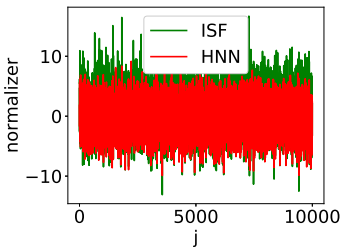


Figure 3: Hubness normalizers in log domain: $\log \sum_i \exp(-D_{i,j}/\epsilon)$ for ISF, and $-\beta_j/\epsilon$ for HNN, which is smoother.

Table 1: Compare NN, ISF and HNN on the simulated retrieval task

	P@1	P@5	P@10
NN	51.71	71.88	78.51
ISF	73.75	88.57	92.36
HNN (Algorithm 1)	**78.85**	**91.43**	**94.60**
HNN (Algorithm 2)	**78.84**	**91.41**	**94.59**

over the iterations in algorithm 1. The accuracy at the first iteration matches that of ISF, validating the comments in remark 1. Next, recall the observation we made in remark 2: Algorithm 2 and ISF both seek normalizers over the target examples to penalize hubness. It is therefore interesting to compare the two normalizers, shown in figure 3. We observe that the normalizer by HNN is smoother than that of ISF.

5.2 BLI Data and Setups

We now compare the different methods on real BLI tasks. We follow the setup in (Conneau et al., 2018). The word embeddings and groundtruth dictionaries (for both training and testing) can be downloaded from the MUSE[4] repository. The dataset includes word embeddings for 45 languages. We focus on six languages in our experiments, since dictionaries are available for any two out of the six. These six languages are English (en), Spanish (es), French (fr), Italian (it), Portuguese (pt) and German (de).

Dictionaries for a pair of languages have the following three parts:

1. src-tgt.0-5000.txt is a **seeding dictionary** for learning the mapping, which has 5K unique source words.

2. src-tgt.5000-6500.txt is a **small test dictionary** that includes 1.5K unique source words. In (Conneau et al., 2018), P@1 values are reported on this set.

3. src-tgt.txt is a full dictionary that includes the above two dictionaries and much more src-to-tgt translations. In later experiments, we will use it as a **large test dictionary** by excluding from it the items in src-tgt.0-5000.txt.

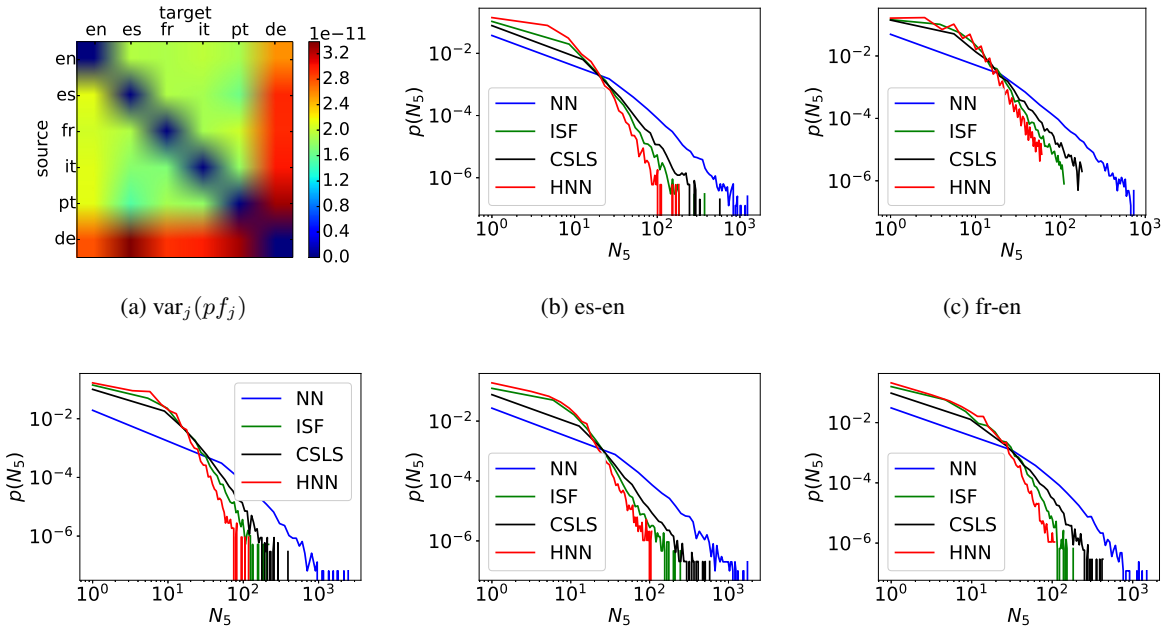
Following (Conneau et al., 2018), an orthonormal mapping $\mathbf{T}$ is learned using the seeding dictionary first. The retrieval step uses cosine distance, *i.e.*,

$$D_{i,j} = \frac{1}{2}\left(1 - \frac{\mathbf{y}_j^\top \mathbf{T}\mathbf{x}_i}{\|\mathbf{x}_i\|\|\mathbf{y}_j\|}\right). \quad (6)$$

The hyper-parameters (ϵ for ISF and HNN, k for CSLS) should be set as the ones that achieve the best accuracy on the seeding dictionary. We choose to trust the default values ($\epsilon = 1/30$ and $k = 10$) used in the MUSE repository, since using them, we can reproduce the results reported in (Conneau et al., 2018). For HNN, ϵ is set to the same value as in ISF, and we use the gradient solver (Algorithm 2) throughout.

As a sanity check, we first reproduce some results reported in Tab. 1 of (Conneau et al., 2018) (the part with cross-lingual supervision), and compare those to HNN. Source and target vocabularies are both 200K. P@1 values are reported on the 1.5K small test dictionary, shown in Tab. 2. HNN is within the ballpark of state-of-the-art, produced by ISF and CSLS.

[4] https://github.com/facebookresearch/MUSE



(a) $\text{var}_j(pf_j)$ (b) es-en (c) fr-en

(d) it-en (e) pt-en (f) de-en

Figure 4: Diagnostics to understand the results in Table 3: (a) Visualizing $\text{var}_j[pf_j]$, for every pair of source and target languages, the corresponding block color-codes the variance of the preferences over the target words. The variances are significantly bigger for pairs that involve German; (b)-(f): Distribution of N_5, the 5-occurrences for different methods in foreign-English BLI tasks. 5-occurrence is the number of times a target word appears in the top-5 retrievals. Long tail of the distribution indicates hubness.

Table 2: Compare HNN with some reproduced results in Tab.1 of (Conneau et al., 2018), using 200K vocabulary. P@1 is reported on the small 1.5K test dictionary. HNN is comparable with ISF and CSLS. Note that the P@1's are reported on the small test dictionary, which is less convincing. For more convincing comparisons, refer to table 3 and table 1 in supplementary.

	en-es	es-en	en-fr	fr-en
NN	77.40	77.27	74.93	76.07
ISF	81.06	82.60	81.07	81.33
CSLS	**81.40**	**82.87**	81.07	**82.40**
HNN	81.27	82.53	**81.33**	82.00

5.3 BLI Results on the Large Test Dictionary

Reporting P@1 on the 1.5K small test dictionary may not be sufficient to make a convincing comparison. We therefore repeat the same experiments but report P@1 on the large test dictionary.

We first keep the vocabulary size of 200K, then try a more challenging 500K. P@1 values are reported on the large test dictionary. In fact, results have the same trend for these different vocabulary sizes. Therefore, considering space limit, we put the results for the 200K case in supplementary. Results of the 500K case are in Tab. 3.

HNN outperforms all the other methods in all cases except pairs that involve German. Note that French, Italian, and Portuguese are all Romance languages. German is a Germanic language. English originates from both. The results seem to suggest that the equal preference assumption is not true between Romance and Germanic languages.

To better understand this, we estimate the "groundtruth" preference of target words, and the variance of the preferences, following the process in section 3.2 (Eq. (2) and (3)). The larger the variance, the less likely the equal preference assumption holds. We visualize the variance between any pair of languages in figure 4a, where a hot block indicates large variance. We observe a large variance when either the source or target language is German. In other words, the equal preference assumption is more violated when translating from or to German.

5.4 Analysis of Hubs in BLI

To see how the hubs are reduced, we again calculate the *k-occurrence* metric. Figure 4b to 4f plot

Table 3: P@1 values on the large test dictionary. Source and target vocabularies are both 500K. HNN is the best except pairs that involve German.

source \ target		en	es	fr	it	pt	de
en	NN		54.98	55.66	46.30	37.02	53.03
	ISF		70.35	72.31	63.75	53.02	65.75
	CSLS		71.21	72.62	64.11	53.54	**66.50**
	HNN		**71.34**	**73.65**	**64.91**	**54.03**	64.61
es	NN	59.87		60.76	61.95	66.94	48.73
	ISF	73.11		76.82	76.33	78.67	61.70
	CSLS	73.02		76.44	76.44	80.29	**62.29**
	HNN	**74.38**		**78.24**	**77.86**	**81.09**	60.78
fr	NN	61.60	61.73		59.43	46.31	57.10
	ISF	74.46	75.72		73.78	60.89	69.07
	CSLS	74.88	76.68		74.34	62.06	**70.34**
	HNN	**75.97**	**77.23**		**75.12**	**63.10**	67.94
it	NN	51.38	64.63	61.45		51.91	50.68
	ISF	65.57	77.76	76.64		67.32	63.58
	CSLS	65.32	78.45	76.74		68.85	**64.57**
	HNN	**67.57**	**79.75**	**78.56**		**70.33**	62.96
pt	NN	42.21	68.93	47.48	50.98		37.95
	ISF	55.76	81.67	64.37	68.37		51.07
	CSLS	54.75	81.98	63.68	67.92		**51.78**
	HNN	**57.43**	**83.97**	**66.19**	**70.44**		49.93
de	NN	56.06	44.33	52.78	45.44	33.20	
	ISF	**69.74**	**60.77**	**71.59**	**65.99**	**52.74**	
	CSLS	68.65	59.21	69.88	63.69	50.72	
	HNN	69.20	60.22	70.71	65.09	52.08	

the distribution of the N_5 values for all methods on foreign-to-English BLI tasks. HNN has the shortest tail in all cases, indicating the fewest hubs.

Table 4: Some representative hubs when applying NN for the pt-en BLI task. Note that the N_5 values are significantly reduced after applying HNN.

word	N_5 by NN	N_5 by HNN	frequency rank
conspersus	1,776	0	484,387
oryzopsis	1,235	5	472,161
these	1,042	25	122
s+bd	912	16	440,835
were	798	24	40
you	474	20	50
would	467	40	73

It is interesting to see what types of words are likely to be hubs. Table 4 lists some representative ones in the pt-en experiment, picked from the top 100 hubs with the biggest N_5 values. Some of them are extremely low-frequency words, e.g., "s+bd". This is consistent with the finding in (Dinu et al., 2014). However, it is also interesting to see that highly frequent words like "were" and "you" also appear to be hubs. Finally, all the N_5 values are reduced after applying HNN.

6 Conclusion

This paper studies how to reduce hubness during retrieval, a crucial step for Bilingual Lexicon Induction (BLI). The Hubless Nearest Neighbor (HNN) is proposed by assuming an "equal preference" constraint. HNN connects to NN, and also sheds light on a recent hubness-preventing method called Inverted SoFtmax (ISF). Empirical results demonstrate that HNN effectively reduces hubs, and can outperform NN, ISF and other state-of-the-art. Future works include applying the method to more language pairs and more domain-specific lexicon induction, e.g., terminologies.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *the 55th Annual Meeting of the Association for Computational Linguistics*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Generalizing and improving bilingual word embedding mappings with a multi-step framework of linear transformations. In *32nd AAAI Conference on Artificial Intelligence*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018b. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *the 56th Annual Meeting of the Association for Computational Linguistics*.

Jean-Julien Aucouturier and Francois Pachet. 2008. A scale-free distribution of false positives for a large class of audio similarity measures. *Pattern recognition*, 41(1):272–284.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

M. Cuturi. 2013. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*.

Georgiana Dinu, Angeliki Lazaridou, and Marco Baroni. 2014. Improving zero-shot learning by mitigating the hubness problem. *arXiv:1412.6568*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *the 14th Conference of the European Chapter of the Association for Computational Linguistics*.

Pascale Fung. 1998. A statistical view on bilingual lexicon extraction: from parallel corpora to non-parallel corpora. In *Association for Machine Translation in the Americas*.

Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. 2016. Stochastic optimization for large-scale optimal transport. In *Advances in neural information processing systems*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*.

Ann Irvine and Chris Callison-Burch. 2017. A comprehensive analysis of bilingual lexicon induction. *Computational Linguistics*, 43(2):273–310.

Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. 2012. Toward statistical machine translation without parallel corpora. In *the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. 2011. Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. In *the 15th conference on computational natural language learning. Association for Computational Linguistics*, pages 154–162.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *33rd Annual Meeting of the Association for Computational Linguistics*.

Samuel L. Smith, David H. P. Turban, Steven Hamblin, and Nils Y. Hammerla. 2017. Offline bilingual word vectors, orthogonal transformations and the inverted softmax. In *International Conference on Learning Representations*.

Ikumi Suzuki, Hara Kazuo, Shimbo Masashi, Saerens Marco, and Fukumizu Kenji. 2013. Centering similarity measures to reduce hubs. In *the 2013 conference on empirical methods in natural language processing*.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. 2015. Normalized word embedding and orthogonal transform for bilingual word translation. In *the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Li Zhang, Tao Xiang, and Shaogang Gong. 2017a. Learning a deep embedding model for zero-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition*.

Meng Zhang, Yang Liu, Huanbo Luan, and Maosong Sun. 2017b. Earth mover's distance minimization for unsupervised bilingual lexicon induction. In *the 2017 Conference on Empirical Methods in Natural Language Processing*.